Appendix (Code)

Sergio Arnaud

10 january, 2019

```r
library(tidyverse)

library(corrplot)
require(gridExtra)

library(caret)
library(xgboost)

library(xgboostExplainer) #install_github("AppliedDataSciencePartners/xgboostExplainer")
```

Reading and exploring the data

Reading data

```r
data <- read_csv('CCPP/data.csv')
```

```
## Parsed with column specification:
## cols(
##   AT = col_double(),
##   V = col_double(),
##   AP = col_double(),
##   RH = col_double(),
##   PE = col_double()
## )
```

```r
summary(data)
```

```
##        AT              V               AP               RH        
##  Min.   : 1.81   Min.   :25.36   Min.   : 992.9   Min.   : 25.56  
##  1st Qu.:13.51   1st Qu.:41.74   1st Qu.:1009.1   1st Qu.: 63.33  
##  Median :20.34   Median :52.08   Median :1012.9   Median : 74.97  
##  Mean   :19.65   Mean   :54.31   Mean   :1013.3   Mean   : 73.31  
##  3rd Qu.:25.72   3rd Qu.:66.54   3rd Qu.:1017.3   3rd Qu.: 84.83  
##  Max.   :37.11   Max.   :81.56   Max.   :1033.3   Max.   :100.16  
##        PE       
##  Min.   :420.3  
##  1st Qu.:439.8  
##  Median :451.6  
##  Mean   :454.4  
##  3rd Qu.:468.4  
##  Max.   :495.8  
```

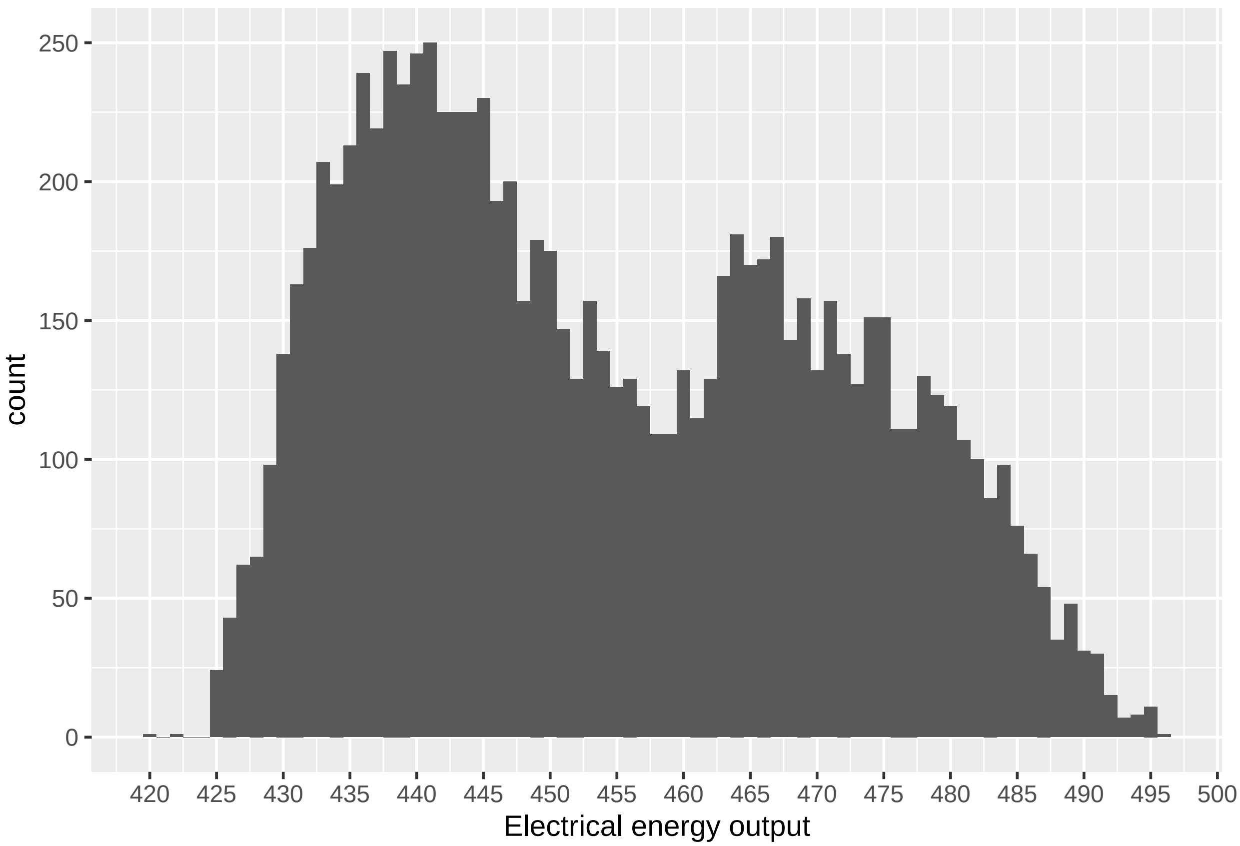
The data dimensions are 9568,5. We have 1231 observations of 5 variables which are: - Temperature (T) - Ambient Pressure (AP) - Relative Humidity (RH) - Exhaust Vacuum (V) - Electrical energy output (EP)

Each variable is continous and the Electrical energy output is the variable to predict.

Variables

The response variable PE, the net hourly electrical energy output of the plant has the following distribution.

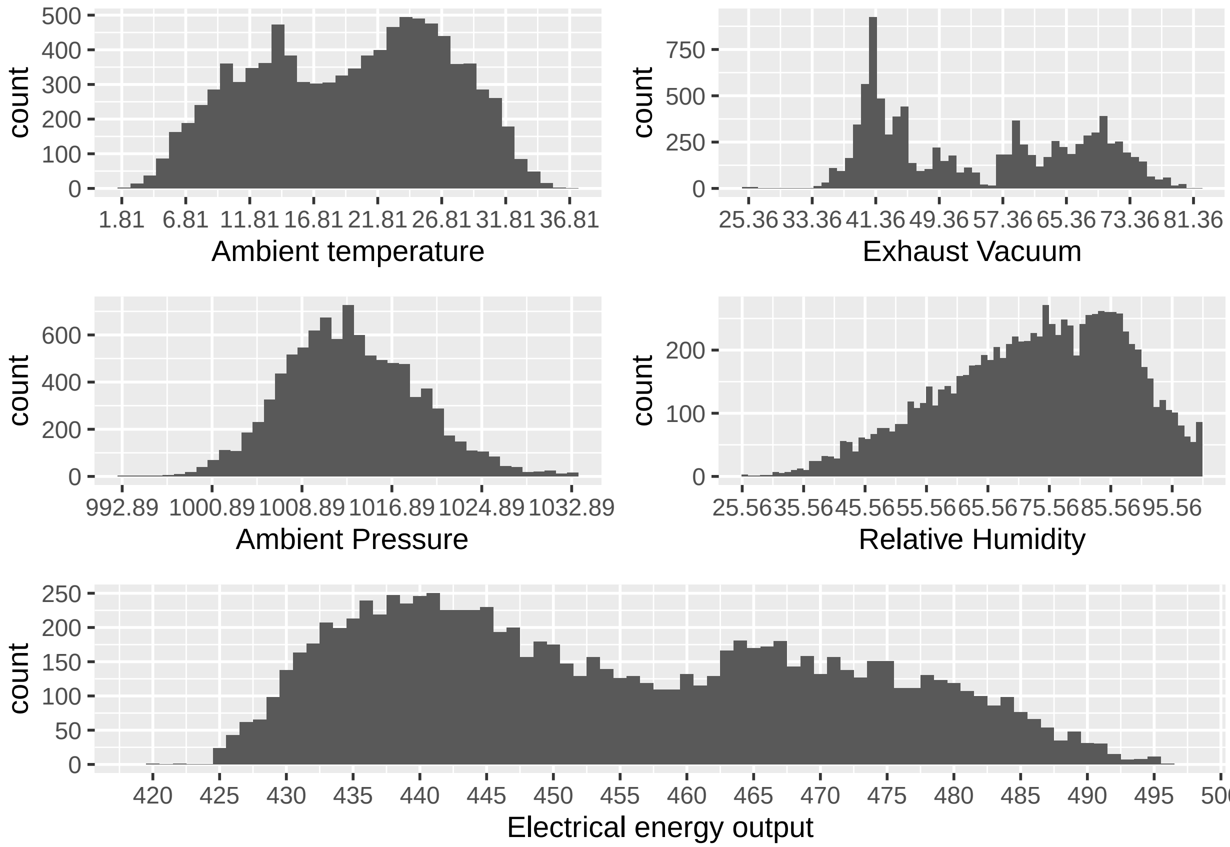
```
ggplot(data=data, aes(x=PE)) +
        geom_histogram(binwidth = 1) +
        scale_x_continuous(breaks= seq(410, 500, by=5)) +
        xlab('Electrical energy output')
```



And the distributions of the 5 variables are

```
p1 <- ggplot(data=data, aes(x=AT)) +
      geom_histogram(binwidth = 1) +
      scale_x_continuous(breaks= seq(min(data$AT), max(data$AT), by=5))+
      xlab('Ambient temperature')

p2 <- ggplot(data=data, aes(x=V)) +
      geom_histogram(binwidth = 1) +
      scale_x_continuous(breaks= seq(min(data$V), max(data$V), by=8))+
      xlab('Exhaust Vacuum')

p3 <- ggplot(data=data, aes(x=AP)) +
      geom_histogram(binwidth = 1) +
      scale_x_continuous(breaks= seq(min(data$AP), max(data$AP), by=8))+
      xlab(' Ambient Pressure')

p4 <- ggplot(data=data, aes(x=RH)) +
      geom_histogram(binwidth = 1) +
      scale_x_continuous(breaks= seq(min(data$RH), max(data$RH), by=10))+
```

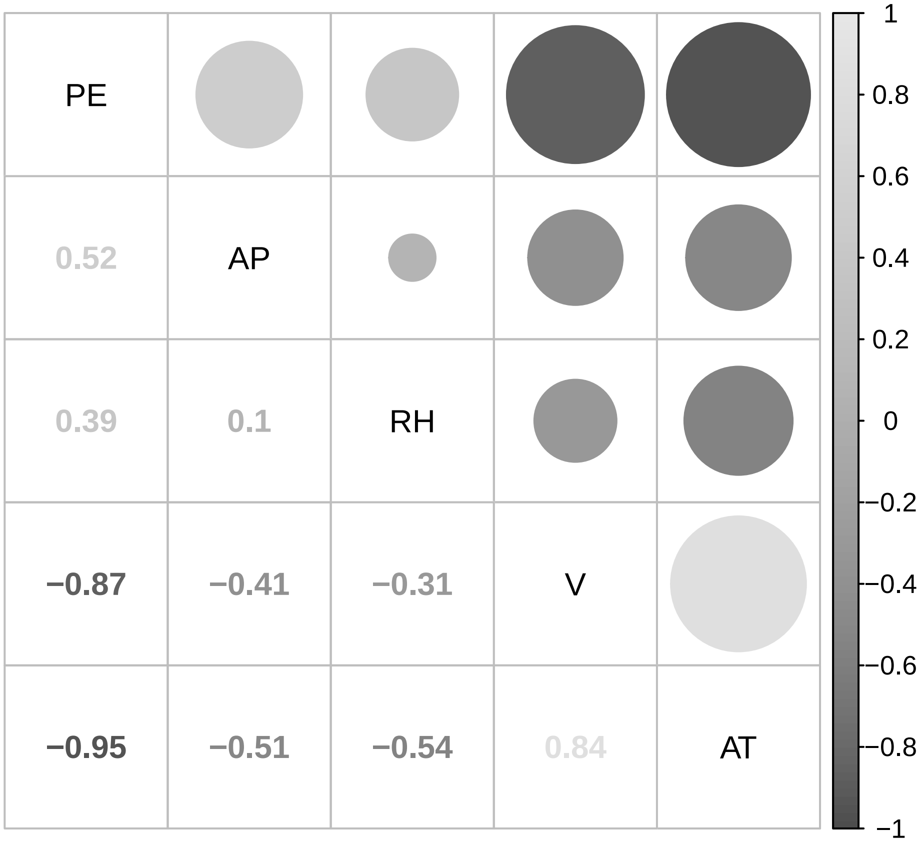
```
      xlab('Relative Humidity')

p5 <- ggplot(data=data, aes(x=PE)) +
        geom_histogram(binwidth = 1) +
        scale_x_continuous(breaks= seq(410, 500, by=5)) +
        xlab('Electrical energy output')

grid.arrange(grobs = list(p1,p2,p3,p4,p5), layout_matrix = rbind(c(1,2),
                                                                  c(3,4),
                                                                  c(5,5)))
```



Relations of the predictors with the response variable

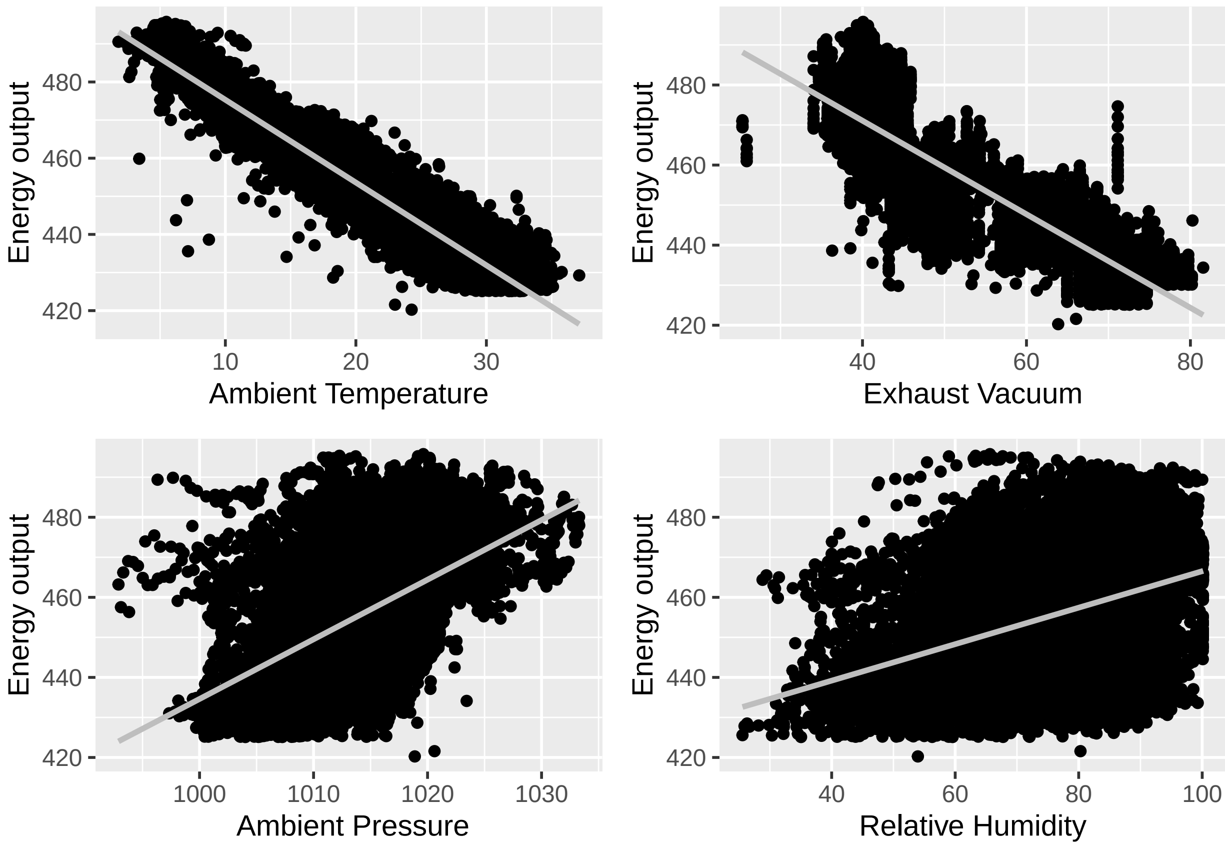
First, let's take a look to the correlation matrix and, in particular, focus in the correlations of each variable with the response variable.

```
correlations <- cor(data, use="pairwise.complete.obs")

cor_sorted <- sort(correlations[,'PE'], decreasing = TRUE)
correlations <- correlations[names(cor_sorted), names(cor_sorted)]

corrplot.mixed(correlations, tl.col="black",
               lower.col = gray.colors(100),
               upper.col = gray.colors(100))
```



And the plots relating each predictor with the response variable

```
p1 <- ggplot(data=data, aes(x=AT, y=PE))+
        geom_point() +
        geom_smooth(method = "lm", se=FALSE, color="grey", aes(group=1)) +
        xlab('Ambient Temperature') +
        ylab('Energy output')

p2 <- ggplot(data=data, aes(x=V, y=PE))+
        geom_point() +
        geom_smooth(method = "lm", se=FALSE, color="grey", aes(group=1)) +
        xlab('Exhaust Vacuum') +
        ylab('Energy output')

p3 <- ggplot(data=data, aes(x=AP, y=PE))+
        geom_point() +
        geom_smooth(method = "lm", se=FALSE, color="grey", aes(group=1)) +
        xlab(' Ambient Pressure') +
        ylab('Energy output')

p4 <- ggplot(data=data, aes(x=RH, y=PE))+
        geom_point() +
        geom_smooth(method = "lm", se=FALSE, color="grey", aes(group=1))+
        xlab('Relative Humidity') +
        ylab('Energy output')

grid.arrange(p1,p2,p3,p4,ncol=2)
```



Preparing data to modelling

Splitting the data in train and test sets

```
set.seed(970628)
trainIndex <- createDataPartition(1:length(data$AT), p=0.8, list=FALSE)

#splitting data into training/testing data using the trainIndex object
train_data <- data[trainIndex,]
test_data <- data[-trainIndex,]

dim(train_data)
```

```
## [1] 7656    5
```

```
dim(test_data)
```

```
## [1] 1912    5
```

```
train_data
```

```
## # A tibble: 7,656 x 5
##        AT     V    AP    RH    PE
##     <dbl> <dbl> <dbl> <dbl> <dbl>
##   1 15.0   41.8 1024.  73.2  463.
##   2 25.2   63.0 1020.  59.1  444.
##   3  5.11  39.4 1012.  92.1  489.
##   4 20.9   57.3 1010.  76.6  446.
##   5 10.8   37.5 1009.  96.6  474.
##   6 26.3   59.4 1012.  58.8  444.
```

```
##  7 15.9   44.0 1014.  75.2  467.
##  8 14.6   45   1022.  41.2  476.
##  9 11.7   43.6 1015.  70.7  478.
## 10 20.1   46.9 1015.  64.2  454.
## # ... with 7,646 more rows
```

It's important to underline that the data have no misssing values and it looks like we don't have to deal with outliers or further cleaning the data, well just standardise each variable.

```
col_mean <- map(train_data, mean)
col_sd <- map(train_data, sd)

train_data <- train_data %>%
    map2_df(col_mean, ~.x - .y) %>% # Remove mean
    map2_df(col_sd, ~.x / .y) # Divide by sd

test_data <- test_data %>%
    map2_df(col_mean, ~.x - .y) %>%
    map2_df(col_sd, ~.x / .y)
```

Linear model

```
lin_model <- lm(data = train_data, PE ~ .)
summary(lin_model)
```

```
##
## Call:
## lm(formula = PE ~ ., data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.54162 -0.18501 -0.00749  0.18557  1.03778
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -3.644e-16  3.038e-03    0.000        1
## AT          -8.596e-01  7.471e-03 -115.053  < 2e-16 ***
## V           -1.766e-01  6.079e-03  -29.057  < 2e-16 ***
## AP           2.459e-02  3.654e-03    6.729 1.83e-11 ***
## RH          -1.347e-01  3.959e-03  -34.020  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2658 on 7651 degrees of freedom
## Multiple R-squared:  0.9294, Adjusted R-squared:  0.9293
## F-statistic: 2.517e+04 on 4 and 7651 DF,  p-value: < 2.2e-16
```

And the mean square error is

```
ans_linear <- predict(lin_model, test_data %>% select(-PE))
ans_linear <- test_data %>% mutate(real = PE, estimated = ans_linear) %>% select(real, estimated)
ans_linear %>% mutate(mse = (real-estimated)^2) %>% summarise(mean(mse))
```

```
## # A tibble: 1 x 1
##   `mean(mse)`
```

```
##         <dbl>
## 1      0.0726
```

Lasso

Cross validation to fix the lambda

```
my_control <- trainControl(method="cv", number=5)
lasso_grid <- expand.grid(
                          alpha = 1,  # alpha = 1 lasso, # alpha = 0 ridge
                          lambda = seq(0.001,1,by = 0.001)
                          )

lasso_model <- train(x = train_data %>% select(-PE),
                     y = train_data$PE,
                     method='glmnet',
                     trControl = my_control,
                     tuneGrid = lasso_grid)
lasso_model$bestTune
```

```
##   alpha lambda
## 2     1  0.002
```

Mean squared erro

```
predictions_lasso <- predict(lasso_model, test_data %>% select(-PE))
ans_lasso <- test_data %>% mutate(real = PE, estimated = predictions_lasso) %>% select(real, estimated)
ans_lasso %>% mutate(mse = (real-estimated)^2) %>% summarise(mean(mse))
```

```
## # A tibble: 1 x 1
##   `mean(mse)`
##         <dbl>
## 1      0.0725
```

XGBoost

Caret hyper parameter tuning

```
xgb_grid = expand.grid(
    nrounds = 1000,
    eta = c(0.3, 0.2, 0.1),
    max_depth = c(3, 4, 5, 6),
    gamma = 0,
    colsample_bytree=1,
    min_child_weight=c(1, 2, 3, 4),
    subsample=1
)

xgb_caret <- train(x = train_data %>% select(-PE),
                     y = train_data$PE,
                     method='xgbTree',
                     trControl = my_control,
                     tuneGrid = xgb_grid)
```

The results are

```r
head(xgb_caret$results)
```

```
##    eta max_depth gamma colsample_bytree min_child_weight subsample nrounds
## 1  0.1         3     0                1                1         1    1000
## 2  0.1         3     0                1                2         1    1000
## 3  0.1         3     0                1                3         1    1000
## 4  0.1         3     0                1                4         1    1000
## 17 0.2         3     0                1                1         1    1000
## 18 0.2         3     0                1                2         1    1000
##         RMSE  Rsquared       MAE      RMSESD  RsquaredSD       MAESD
## 1  0.1902446 0.9638717 0.1374837 0.003953519 0.002244040 0.001313633
## 2  0.1897812 0.9640582 0.1375092 0.003958975 0.002161389 0.001028581
## 3  0.1901313 0.9639191 0.1377980 0.004045142 0.002257899 0.002071299
## 4  0.1909614 0.9636070 0.1387045 0.003813614 0.002164421 0.002170217
## 17 0.1845860 0.9659834 0.1306439 0.003648292 0.001934761 0.001810661
## 18 0.1838510 0.9662304 0.1309183 0.005428448 0.002561988 0.002259691
```

And the best tune for the hyperparameters is

```r
xgb_caret$bestTune
```

```
##    nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 13    1000         6 0.1     0                1                1         1
```

Now let us find the ideal number of rounds

```r
label_train <- train_data$PE

# put our testing & training data into two seperates Dmatrixs objects
dtrain <- xgb.DMatrix(data = as.matrix(train_data %>% select(-PE)), label= label_train)
dtest <- xgb.DMatrix(data = as.matrix(test_data %>% select(-PE)))

default_param<-list(
        objective = "reg:squarederror",
        booster = "gbtree",
        eta=0.1,
        gamma=0,
        max_depth=6,
        min_child_weight=2
)

xgbcv <- xgb.cv( params = default_param, data = dtrain, nrounds = 1000, nfold = 5, showsd = T, stratifi
```

```
## [1]  train-rmse:1.011927+0.002215    test-rmse:1.012305+0.009122
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [41] train-rmse:0.180086+0.002055    test-rmse:0.212726+0.007439
## [81] train-rmse:0.159986+0.001998    test-rmse:0.202507+0.007983
## [121]    train-rmse:0.145730+0.001242    test-rmse:0.195676+0.008700
## [161]    train-rmse:0.133892+0.001974    test-rmse:0.190461+0.008799
## [201]    train-rmse:0.124802+0.001873    test-rmse:0.187532+0.008928
## [241]    train-rmse:0.117295+0.001391    test-rmse:0.185282+0.009086
## [281]    train-rmse:0.110211+0.001545    test-rmse:0.183131+0.009027
## [321]    train-rmse:0.103899+0.001570    test-rmse:0.181547+0.009086
## [361]    train-rmse:0.098591+0.001698    test-rmse:0.180629+0.009304
```

```
## [401]    train-rmse:0.093530+0.001621    test-rmse:0.179902+0.009647
## [441]    train-rmse:0.088497+0.001299    test-rmse:0.178979+0.009841
## [481]    train-rmse:0.084607+0.001161    test-rmse:0.178477+0.009879
## [521]    train-rmse:0.080424+0.000982    test-rmse:0.177907+0.009813
## [561]    train-rmse:0.076793+0.000866    test-rmse:0.177613+0.009912
## [601]    train-rmse:0.073512+0.000792    test-rmse:0.177261+0.010111
## Stopping. Best iteration:
## [628]    train-rmse:0.071292+0.000912    test-rmse:0.177089+0.010107
```

The optimal number of rounds for the choice of hyper parameters is 598. Training the model with the optimal hyperparameters

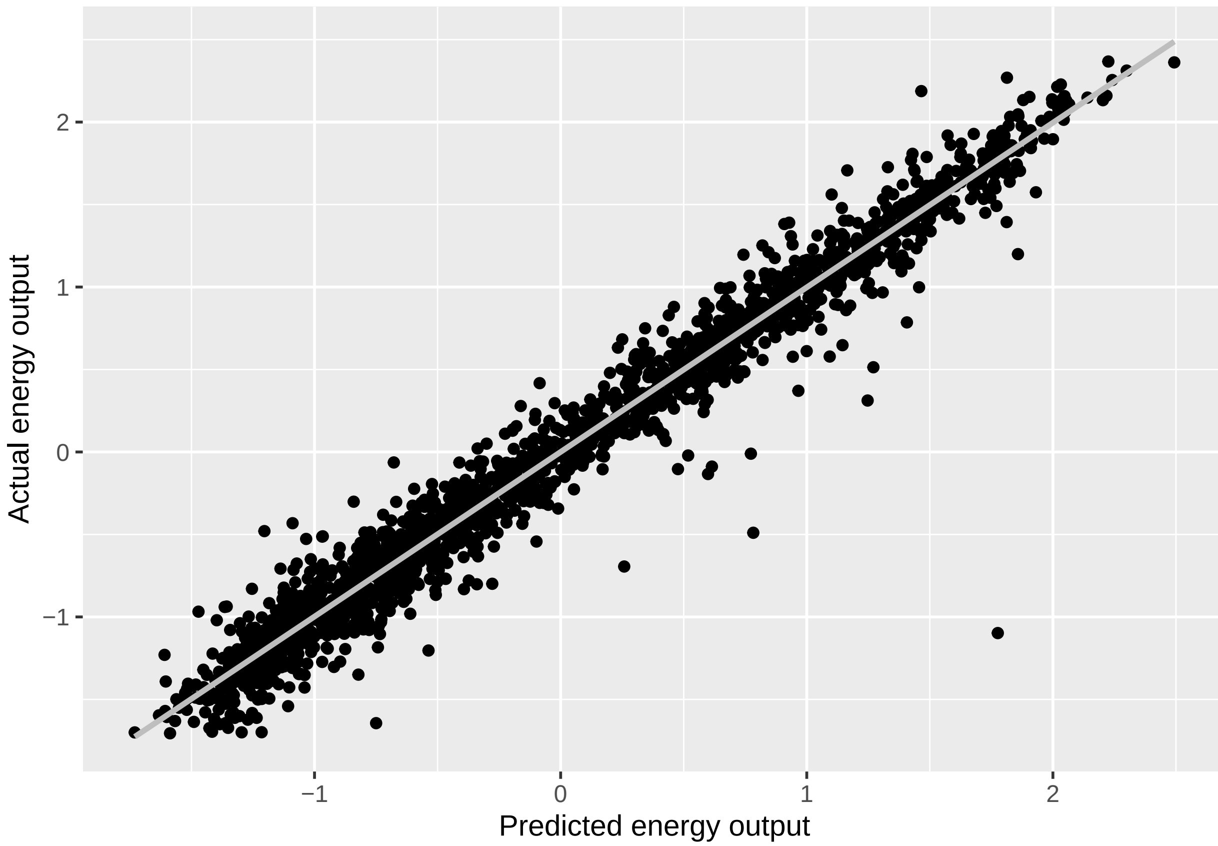
```
xgb_mod <- xgb.train(data = dtrain, params=default_param, nrounds = 598)
```

Predicting and evaluating the prediction results with the test dataset

```
XGBpred <- predict(xgb_mod, dtest)
ans_xgb <- test_data %>% mutate(real = PE, estimated = XGBpred) %>% select(real, estimated)
ans_xgb %>% mutate(mse = (real-estimated)^2) %>% summarise(mean(mse))
```

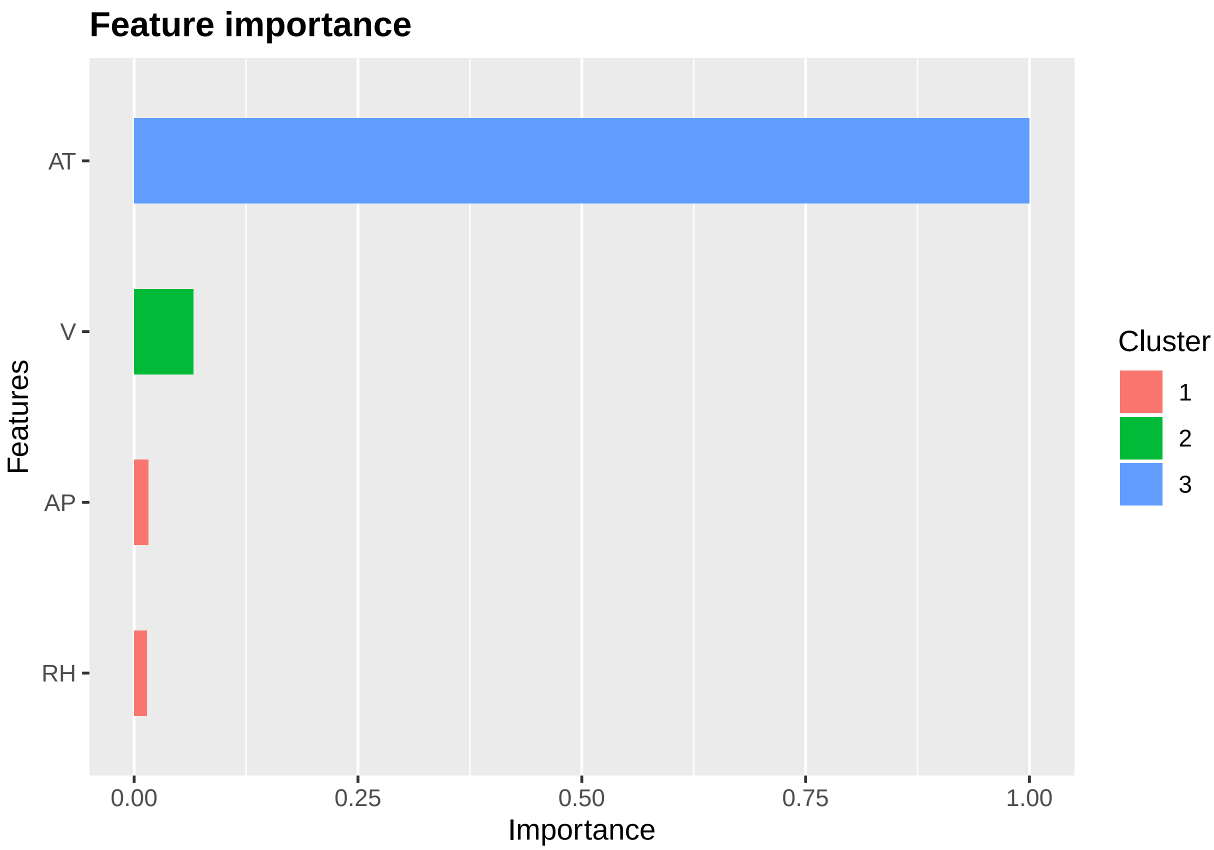
```
## # A tibble: 1 x 1
##   `mean(mse)`
##         <dbl>
## 1      0.0324
```

```
ggplot(data=ans_xgb, aes(x=estimated, y=real))+
        geom_point() +
        geom_smooth(method = "lm", se=FALSE, color="grey", aes(group=1)) +
        ylab('Actual energy output') +
        xlab('Predicted energy output')
```



Plotting the variable importance

```
library(Ckmeans.1d.dp) #required for ggplot clustering
mat <- xgb.importance (feature_names = colnames(train_data),model = xgb_mod)
xgb.ggplot.importance(importance_matrix = mat, rel_to_first = TRUE)
```



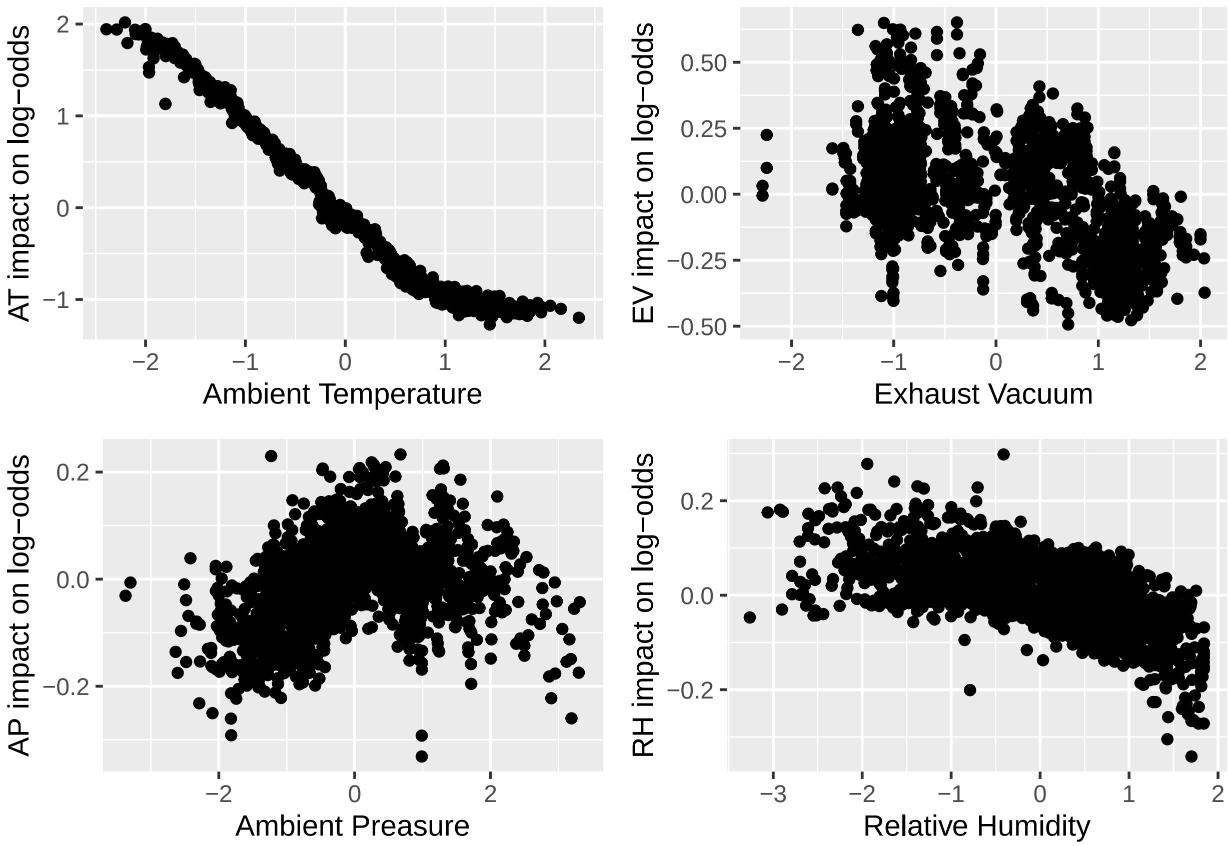
Building explainer of results

```
explainer = buildExplainer(xgb_mod, dtrain, type="binary", base_score = 0.5, trees_idx = NULL)
pred.breakdown = explainPredictions(xgb_mod, explainer, dtest)
colnames(pred.breakdown) <- paste("pred", colnames(pred.breakdown), sep = "_")
```

Each plot in the following figure shows the values of a feature plotted against the impact associated with that value.

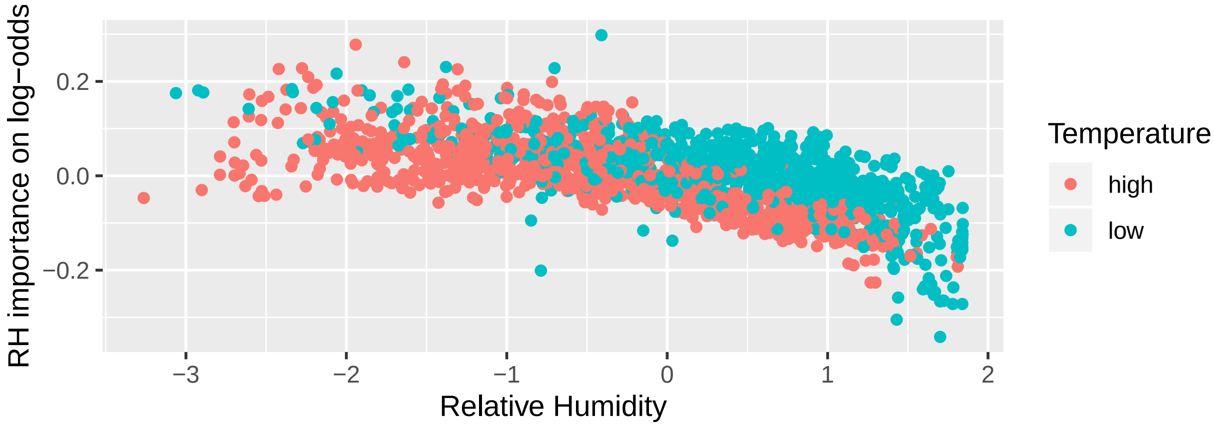
```
expl_data <- cbind(test_data, pred.breakdown)
expl_data <- expl_data %>% mutate(Temperature = ifelse(AT > 0, "high", "low"))

p1 <- ggplot(data=expl_data, aes(x=AT, y=pred_AT))+
        geom_point() +
        xlab('Ambient Temperature') +
        ylab('AT impact on log-odds')

p2 <- ggplot(data=expl_data, aes(x=V, y=pred_V))+
        geom_point()  +
        xlab('Exhaust Vacuum') +
        ylab('EV impact on log-odds')

p3 <- ggplot(data=expl_data, aes(x=AP, y=pred_AP))+
        geom_point() +
        xlab('Ambient Preasure') +
        ylab('AP impact on log-odds')

p4 <- ggplot(data=expl_data, aes(x=RH, y=pred_RH))+
```

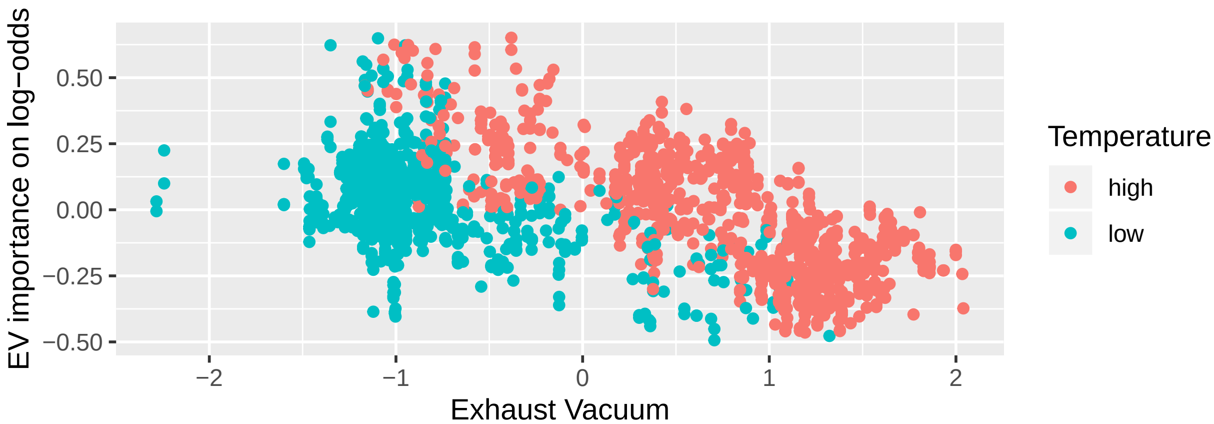
```r
        geom_point() +
        xlab('Relative Humidity') +
        ylab('RH impact on log-odds')

grid.arrange(p1,p2,p3,p4,ncol=2)
```



Each plot in the following figure shows the values of a feature plotted agains the impact associated with that value. Colored by low or high temperature.

```r
p1 <- ggplot(data=expl_data, aes(x=V, y=pred_V, col = Temperature))+
        geom_point() +
        xlab('Exhaust Vacuum') +
        ylab('EV importance on log-odds')

p2 <- ggplot(data=expl_data, aes(x=RH, y=pred_RH, col = Temperature))+
        geom_point() +
        xlab('Relative Humidity') +
        ylab('RH importance on log-odds')

grid.arrange(p2,p1,ncol=1)
```





Ilustrating how a single observation is built

```r
showWaterfall(xgb_mod, explainer, dtest, test_data,  8, type = "binary")
```

```
##
##
## Extracting the breakdown of each prediction...
##
  |
  |                                                                      |   0%
  |
  |                                                                      |   1%
  |
  |=                                                                     |   1%
  |
  |=                                                                     |   2%
  |
  |==                                                                    |   2%
  |
  |==                                                                    |   3%
  |
  |==                                                                    |   4%
  |
  |===                                                                   |   4%
  |
  |===                                                                   |   5%
  |
```

```
|====                                                                  |   5%
|
|====                                                                  |   6%
|
|=====                                                                 |   7%
|
|=====                                                                 |   8%
|
|======                                                                |   8%
|
|======                                                                |   9%
|
|=======                                                               |   9%
|
|=======                                                               |  10%
|
|=======                                                               |  11%
|
|========                                                              |  11%
|
|========                                                              |  12%
|
|=========                                                             |  12%
|
|=========                                                             |  13%
|
|=========                                                             |  14%
|
|==========                                                            |  14%
|
|==========                                                            |  15%
|
|===========                                                           |  15%
|
|===========                                                           |  16%
|
|============                                                          |  17%
|
|============                                                          |  18%
|
|=============                                                         |  18%
|
|=============                                                         |  19%
|
|==============                                                        |  19%
|
|==============                                                        |  20%
|
|==============                                                        |  21%
|
|===============                                                       |  21%
|
|===============                                                       |  22%
|
```

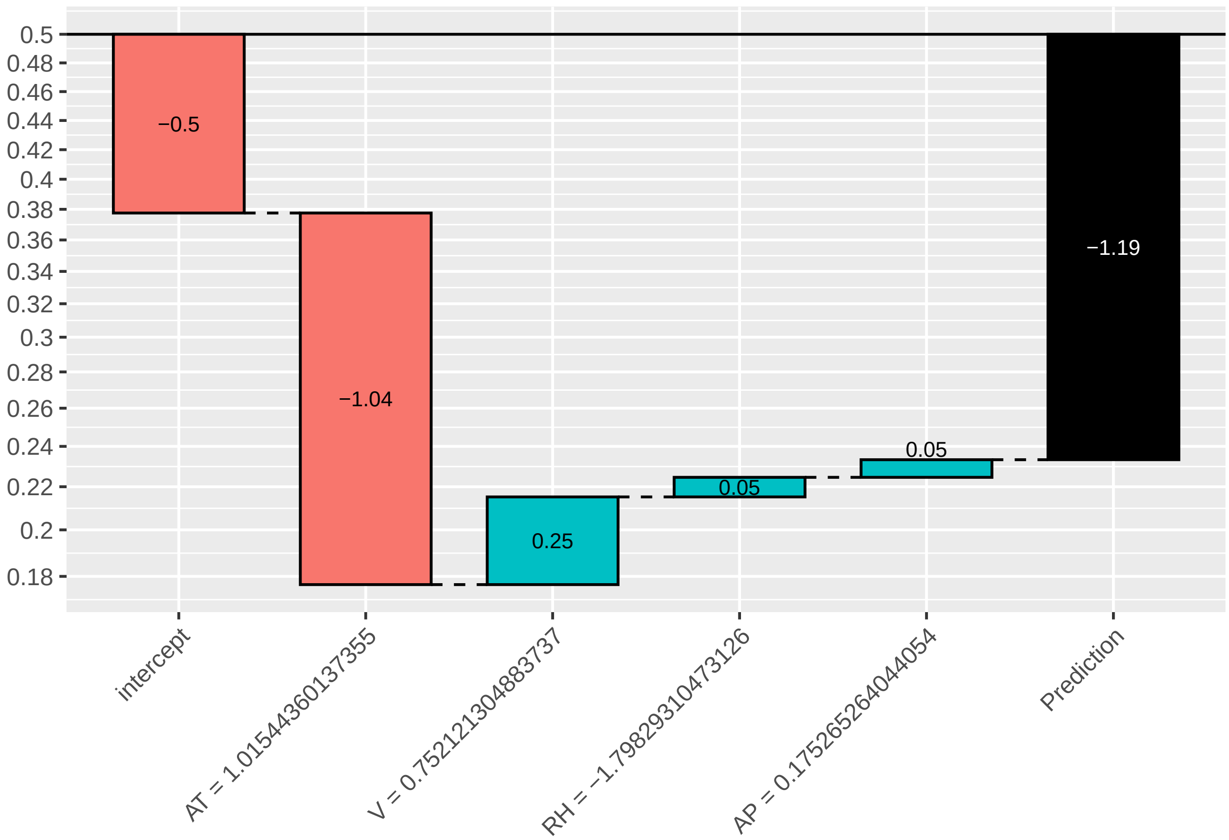
```
|================                                                                      |  22%
|
|================                                                                      |  23%
|
|=================                                                                     |  24%
|
|=================                                                                     |  25%
|
|==================                                                                    |  25%
|
|==================                                                                    |  26%
|
|===================                                                                   |  27%
|
|===================                                                                   |  28%
|
|====================                                                                  |  28%
|
|====================                                                                  |  29%
|
|=====================                                                                 |  29%
|
|=====================                                                                 |  30%
|
|=====================                                                                 |  31%
|
|======================                                                                |  31%
|
|======================                                                                |  32%
|
|=======================                                                               |  32%
|
|=======================                                                               |  33%
|
|========================                                                              |  34%
|
|========================                                                              |  35%
|
|=========================                                                             |  35%
|
|=========================                                                             |  36%
|
|==========================                                                            |  36%
|
|==========================                                                            |  37%
|
|==========================                                                            |  38%
|
|===========================                                                           |  38%
|
|===========================                                                           |  39%
|
|============================                                                          |  39%
|
```

```
|============================                                          |  40%
|
|============================                                          |  41%
|
|=============================                                         |  41%
|
|=============================                                         |  42%
|
|==============================                                        |  42%
|
|==============================                                        |  43%
|
|===============================                                       |  44%
|
|===============================                                       |  45%
|
|================================                                      |  45%
|
|================================                                      |  46%
|
|=================================                                     |  46%
|
|=================================                                     |  47%
|
|=================================                                     |  48%
|
|==================================                                    |  48%
|
|==================================                                    |  49%
|
|===================================                                   |  49%
|
|===================================                                   |  50%
|
|===================================                                   |  51%
|
|====================================                                  |  51%
|
|====================================                                  |  52%
|
|=====================================                                 |  52%
|
|=====================================                                 |  53%
|
|=====================================                                 |  54%
|
|======================================                                |  54%
|
|======================================                                |  55%
|
|=======================================                               |  55%
|
|=======================================                               |  56%
|
```

```
|========================================                              |  57%
|
|========================================                              |  58%
|
|=========================================                             |  58%
|
|=========================================                             |  59%
|
|==========================================                            |  59%
|
|==========================================                            |  60%
|
|==========================================                            |  61%
|
|===========================================                           |  61%
|
|===========================================                           |  62%
|
|============================================                          |  62%
|
|============================================                          |  63%
|
|============================================                          |  64%
|
|=============================================                         |  64%
|
|=============================================                         |  65%
|
|==============================================                        |  65%
|
|==============================================                        |  66%
|
|===============================================                       |  67%
|
|===============================================                       |  68%
|
|================================================                      |  68%
|
|================================================                      |  69%
|
|=================================================                     |  69%
|
|=================================================                     |  70%
|
|=================================================                     |  71%
|
|==================================================                    |  71%
|
|==================================================                    |  72%
|
|===================================================                   |  72%
|
|===================================================                   |  73%
|
```

```
|====================================================                  |  74%
|
|====================================================                  |  75%
|
|=====================================================                 |  75%
|
|=====================================================                 |  76%
|
|======================================================                |  77%
|
|======================================================                |  78%
|
|=======================================================               |  78%
|
|=======================================================               |  79%
|
|========================================================              |  79%
|
|========================================================              |  80%
|
|========================================================              |  81%
|
|=========================================================             |  81%
|
|=========================================================             |  82%
|
|==========================================================            |  82%
|
|==========================================================            |  83%
|
|===========================================================           |  84%
|
|===========================================================           |  85%
|
|============================================================          |  85%
|
|============================================================          |  86%
|
|=============================================================         |  86%
|
|=============================================================         |  87%
|
|=============================================================         |  88%
|
|==============================================================        |  88%
|
|==============================================================        |  89%
|
|===============================================================       |  89%
|
|===============================================================       |  90%
|
|===============================================================       |  91%
|
```

```
  |================================================================      |  91%
  |
  |================================================================      |  92%
  |
  |=================================================================     |  92%
  |
  |=================================================================     |  93%
  |
  |==================================================================    |  94%
  |
  |==================================================================    |  95%
  |
  |===================================================================   |  95%
  |
  |===================================================================   |  96%
  |
  |====================================================================  |  96%
  |
  |====================================================================  |  97%
  |
  |====================================================================  |  98%
  |
  |===================================================================== |  98%
  |
  |===================================================================== |  99%
  |
  |======================================================================|  99%
  |
  |======================================================================| 100%
##
## DONE!
##
## Prediction:  0.2332554
## Weight:  -1.19002
## Breakdown
##   intercept          AT           V          RH          AP
## -0.49999264 -1.03940021  0.24528357  0.05478898  0.04930068
```


0.5
0.48
0.46
0.44
0.42
0.4
0.38
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
−0.5
−1.04
0.25
0.05
0.05
−1.19
intercept
AT = 1.01544360137355
V = 0.752121304883737
RH = −1.79829310473126
AP = 0.17526526404054
Prediction